LINKWINDS, A SYSTEM FOR INTERACTIVE SCIENTIFIC DATA ANALYSIS AND VISUALIZATION

Allan S, Jacobson, Andrew L. Berkin and Martin N. Orton

ABSTRACT

The Linked Windows Interactive Data System (LinkWinds) is a prototype visual data exploration system resulting from a NASA/JPL program of research into graphical methods for rapidly and interactively accessing, displaying and analyzing large multivariate multidisciplinary data sets. Running under UNIX, it is an integrated multi-application execution environment allowing, the dynamic interconnection, through a data-linking paradigm, of multiple windows containing visual displays and controls. This paradigm, which results in a system which functions much like a graphical spreadsheet, is not only a powerful method for organizing, large amounts of data for analysis, but provides a highly intuitive, easy-to-lcwn user interface on top of the traditional graphical user interface. The linking of data displays and controls for their manipulation provides great flexibility in rapidly exploring large masses of complex data to detect trends, correlations and anomalies. The system, containing an expanding suite of non-domain specific applications, provides for the ingestion of a variety of database formats, and hard copy output of all displays. Networked workstations running LinkWinds may be linked, providing a Multi-User Science Environment (MUSE) for collaborative data exploration by geographically separated scientists. The system is being developed in close collaboration with investigators in oceanography, geology, upper atmospheric science and magnetospheric physics. in addition to gridded data sets, LinkWinds has the ability to ingest and display real time data, which may be from spacecraft, laboratory experiments, or computer simulations.

INTRODUCTION

Great advances in sensor and computer technologies, coupled with the ability to transport sensors to remote locations such as deep in the oceans 01 high into space have resulted in an ever increasing flood of data, which, to be useful, must be transported, stored and analyzed in a timely manner. observational systems, computations and computer simulations are being applied in almost all fields of human endeavor in efforts to understand ourselves and our world. in this paper, we report on a system which grew out of a NASA research project at the Jet l'repulsion Laboratory to study the application of computer graphics to the problems of rapidly and interactively exploring and analyzing very large amounts of scientific data gathered either during the execution of observations, computations or computer simulations. The objectives of the research program are (1) to develop a software environment and test bed which will support the rapid prototyping of visual data analysis applications, while at the same time maintaining the high level of performance necessary for interactive manipulation of graphical displays; (?) to develop a user interface that is truly intuitive, allowing quick access to the software for the novice as well as the advanced user; (3) to provide a suite of sample applications which are useful across a variety of scientific disciplines; and (4) to provide tools to support user development of applications for this environment.

LINKWINDS

The Linked Windows Interactive Data System, or LinkWinds, is both a test bed for the concepts resulting from this research and a prototype product of the effort. It is a visual data analysis and exploration system designed to rapidly and interactively investigate large multivariate and multidisciplinary data sets to detect trends, correlations and anomalies. LinkWinds provides a variety of functions and services, including 2-dimensional and 3-dimensional graphical displays of data, hard copy of graphical displays and text, interactive color manipulation, animation creation and display, data subsetting either at the input or output, a journal and macro capability, cxmtcxt-sensitive help, and network support for collaborative data analysis. It is an integrated multi-application execution environment with a full graphical user interface (GUI) running under UNIX. While its GUI is based upon X Windows, LinkWinds draws upon the Silicon Graphics Inc. (SGI) GL-library for rendering support and therefore, presently runs only on workstations supporting that library, This includes all SGI workstations, and those of other manufacturers who have licensed and support the GL-library.

To understand where LinkWinds fits into the array of available tools, it is instructive to divide data visualization support software into four categories. With several noteworthy examples, these are:

a. individual scientific applications such as MacSpin [2].
b. Multi-application execution environments such as Spyglass Transform [1'2].
c. Procedural programming languages such as IDL [16] and PV-Wave [6].
d. Visual programming or data-flow languages such as Apc [3], AVS [13], and Khoros [8].

LinkWinds, in a manner similar to ConMan [4], allows the user to interactively link objects on the workstation screen to accomplish data analysis tasks. While such linking is reminiscent of the programing of data-flow systems, the similarity is only superficial. LinkWinds is a multi-application execution environment, This distinction enables an architectural design which results in more efficient use of memory, optimization for execution, rather than for programming, and a less complex and therefore more easily learned system of operations.

LinkWinds' use as a test bed for visual data analysis research and its application to scientific problems since its inception has resulted in a system which has grown primarily through evolutionary processes. The evolution has been greatly facilitated by its original layout. Although developed in ANSI C rather than an object-oriented language, its foundation and subsequent development have been on an object-orientcxi model. 'There are currently three levels of the software. The application level has one source code module for each of the LinkWinds applications. Header files contain declarations of the methods and a data structure which is kept hidden. Each module has procedures analogous to creators and destructors. The system level contains the Link Winds gadget set, the data manager, the file manager, the Lynx language interface (see below) and utility routines. Finally, the windowing level controls the users' interactions with the applications, the system events, menus, inter- and intra-object messages and the graphics

in addition to a top-level menu, there are three types of objects presented on the screen. They are data, control, and display objects. Each occupies a window on the LinkWinds screen, and communicates with other objects through a message passing protocol. The communication paths are interactively established by the user through a "data-linking" paradigm which will be discussed below. A data object, shown in Figure 1, can be distinguished by its single "linking" button. It has a short form, shown on the left, and a long form show on the right. The expanded form is seen by pressing the "more" button on the short form. Control objects such as the slider in Figure 2 are distinguished by having both a "linking" button and an "unlinking" button. An example of a display object is shown in Figure 3. About half of the current LinkWinds displays also have control functions. The image display show is such an object, and therefore has the "link" and "unlink" buttons. These buttons are missing on pure display objects.

All mcs.sages generated by LinkWinds objects, as well as menu and button selections internal to objects, are recorded as program statements in an underlying language called lynx. The Lynx program statements are the basis for several key LinkWinds functions. The first is the maintenance of an internal journal of all user originated commands executed by the environment. The journal is continuously maintained during a session and saved at its conclusion as a text file. Intermediate files can be saved at any time through a menu option, A file can then be edited for replay, or replayed in its entirety at the initiation of subsequent Link Winds sessions, allowing the user to draw upon a previous layout of 1 ink Winds applications and links, or repeat a full analysis session.

Lynx is also the basis for a macro capability. Through a top-level menu option, the user can turn on a "learn" mode in which all subsequent Lynx program statements are saved, Upon ending the "learn" mode, the file is assigned a name and appears immediately as an item in the "macro" option of the top-level menu for execution at any time.

A third function bawd upon the Lynx program statement and message passing protocol is the LinkWinds Multi-User Science Environment (MUSE), a cooperative work capability which provides a method for multiple executing LinkWinds systems to communicate with one another, Using menu options, users remotely separated can connect to one another over the network, to receive Lynx messages, send them or do both. By simultaneously establishing a telephone voice connection, scientists can discuss their activities while cooperatively viewing and manipulating, their data, A success.ful connection require.s that each user be executing LinkWinds and that each has access to the data sets being analyzed. This is normally arranged by transporting the data sets prior to the collaborative session. The MUSE capability is also very useful in accommodating many of the needs of new users. It is used to give tutorials over the network and to allow users to demonstrate recommendations for application changes or to point out bugs. MUSE is implemented in such a way as to transmit only individual control values, and button or menu picks, and therefore requires virtually no network bandwidth.

Hard copy of the LinkWinds graphical displays arc provided by function keys. Placing the cursor in a window and pressing N produces an image of a window's graphical data display; pressing F2 saves the complete window and frame; and F3 saves the full screen. The figures show were obtained in this manner. Text files containing the results of various statistical operations, and data table listings arc also available via buttons on the relevant applications.

DATA-LINKING AND THE USER INTERFACE

In developing an intuitive and easy-to-lcarm and usc interface, we were guided by several principles [9]. These arc that users, when presented with a new piece of software, arc impatient and want to get started on productive work as quickly as possible; that they will tend to learn from exploring the software rather than reading manuals; and that as long as the software conforms to their expectations, they will make quick progress in learning to usc it, referring to manuals only when specific problems arc encountered.

The LinkWinds data-linking paradigm is onc of its most distinguishing features, and evolved from the need to rapidly and simultaneously explore data from various sources to detect possible relationships, and a desire to create a truly easy-to-lcara and intuitive user interface. Data-linking provides the user the ability to interactively link the many applications of LinkWinds for concerted actions in examining data, and results in systems of applications whose ensembles arc of much greater utility and power than their individual component applications. The data-linking characteristics of spreadsheets, in which cells containing numbers can be thought of as linked to other related cells, provide a powerful way of organizing data for analysis and a natural and intuitive interface. LinkWinds' scheme of data-linking can most easily be understood this context. Formulae arc associated with each spreadsheet cell, so that when a number changes, all cells linked to the changed ccl] recalculate their values. LinkWinds dots the same thing, but in a graphics environment where the rigid grid structure of a spreadsheet gives way to free form, and a cell can translate from a single number into objects such as a slider producing a single value, or images containing large arrays of numbers.

The operation of data-linking is affected through two icons. The link icon is a button displaying two interlocked rings, while the unlink icon displays two rings that arc separated. As mentioned above, LinkWinds' objects may have a single link button, as in the case of the data object shown in Figure 1; the full set of link and unlink buttons, as is the case for all controls such as the slider shown in figure 2 and the display object in figure 3; or-no buttons, as in the case of the pure display objects such as the Globe seen in the lower right corner of Figure 4. To perform a link, a left-mouse button is pressed while the cursor is placed on the appropriate button, and a "rubber band" is dragged out and dropped into the application to be linked. To break the link, the same thing is done using the unlink button. The rubber bands indicating the links may be displayed at any time during a session by placing the cursor in the top-level logo and pressing the left mouse button. They are shown eminating from the source button to the center of the destination object,

There arc two simple rules to follow in applying the linking paradigm:

1. When, as a result of a top-level menu selection of a control or display, an empty window appears on the screen, put data into it. This is done by linking a data object (see Figure 1) into the window. Since data ingested by LinkWinds is in shared memory, no data flows into the newly activated application, Instead, pointers to the data and metadata arc passed providing full access to all the information needed to establish its identity and pa-form its functions.

2. When an object with the pair of link symbols appears, exercise its control function by linking it into the object to be controlled. Once the link is made, the message flow to the object is enabled. For instance, in the case of

the slider in Figure 2, each time the slider is moved, its final value is broadcast to all objects to which it is linked. When the "track" toggle is pressed, a continuous stream of messages results as the slider is moved.

Use of the GUI and the data-linking paradigm results in a user inter'face which is uniform across all of the LinkWinds applications. It is a "direct manipulation" interface [11] in that there is a continuous representation of displays and controls during a session; the system is run by physical actions rather than a complex syntax; there is rapid and immediately visible response to all actions; and all actions are reversible. As a result, new users pick up the operational principles of Link Winds very rapidly through demonstrat ions by experienced users. Our typical experience is that after about thirty minutes of instruction, the new user can begin to carry out productive work. Once the operational concepts are learned, they are retained and subsequent learning is the result of explorations. Retention is quite important for scientists who spend much of their time gathering data rather than analyzing it. The time required to relearn their analysis software must be minimal. Users demonstrate less anxiety in interacting with the system because the responses are predictable and easily rever std. If the direction taken by an analyst is found to be counterproductive, it is easily changed.

DATABASE INTERFACE

The current version of LinkWinds works with both archived and real time data, The archived data mode accepts data in a variety of standard formats. These are:

1. Raw binary data in either byte or float ing point for mat. The data must be arranged as a single image, or, in the case of a 3-dimensional data set, a sequence of images, each in scan line order, Since, in this case, there is no header information, an ancillary data description file, discussed below, must be provided.

2.. The Silicon Graphics, Inc. native RGB format.

3. The Hierarchical Data Format (HDF) [1] created by the National Center for Supercomputing Applications.

4. The Common Data Format (CDF) [1] originated at NASA's Goddard Space Flight Center.

5. The Planetary Data System (PDS) format originated at NASA/JPL.

6. The Flexible Image Transport System (FITS) [15] widely used by the astrophysics community.

Additional data formats are imported using DataHub [5], a system being developed at JPL in close collaboration with the LinkWinds effort. It is currently used to provide Link Winds direct access to a wide variety of data formats including the Airborne Visible and Infrared Imaging Spectrometer (AVIRIS) format, and NetCDF [1][10], a derivative of CDF. DataHub also provide.s a variety of modes for subsetting large data sets.

Data ingestion is controlled by three text files generated by the user. Sample versions of these files are provided as self-explanatory templates. The first is a LinkWinds configuration file. It contains a list of paths to any directories which contain the user's data, The second resides in one of the paths and lists either the user's data description files or data files. These filenames can contain wild cards, All names listed in this text file appear in the top-level menu "Databases" option. The third file contains data description, such as the filenames of all data sets to be associated during the analysis, the number of axes and their names, and other metadata needed to translate data and axis values to numbers meaningful to the user. The data description file can be eliminated if there is adequate metadata in the data file header, which is possible with most of the standard data formats, Each data file listed in these descriptor files appears in the top-level "Data" menu. A color palette may be included in either the data file header or user-defined in the data description file.

The real time mode of LinkWinds is a recent development, and is exercised through the creation of a server tailored to the format of the data stream. The data description file contains the name of the server, as well as the usual auxiliary metadata. The server is connected to LinkWinds through UNIX sockets. An interrupt system within LinkWinds senses the arrival of a data packet and notifies the relevant applications so they can update appropriately. In addition to the incoming measured or simulated data, clock times and engineering information about the status of the sending devices may be read. Incoming data is saved in HDF 8-bit format, and may be viewed later by requesting a replay server designated in the metadata file. Specific data files and time intervals may be chosen for replay, As a proof-of-concept demonstration, five LinkWinds applications were built and used to monitor and interactively analyze real time data from the University of Iowa's Plasma Wave Subsystem aboard the Galileo spacecraft during its second 1 ;arth encounter in December 1992.

Operational ionally, data arc brought into a LinkWinds session through the usc of the data object show in Figure 1. The object represents both the data file and any associated metadata, the most pertinent of which is displayed in the expanded form. in those cases where the data file is too large to be accommodated by the workstation memory and swap space, subsetting can be accomplished using the sliders at the bottom of the expanded form. For instance, the water vapor data set represented by the object in Figure 1 is a sequence of twelve monthly images. The user has decided, in this case, to load only the images for months 4 through month 10. Such decisions can be made either upon the input of the data, or subsequently during the analysis session. In the latter case, all applications using the data will be instantly resized and redrawn and internal memory allocations readjusted. Strategies for subsetting will cent inue to evolve, driven by the very large size of some data files. For instance, a single AVIRIS image typically measures 614 by 512 pixels with 224 16-bit data channels per pixel for a total image size of about 140 MBytes.

APPLICATIONS OVERVIEW

To avoid creating solutions in search of problems, LinkWinds applications arc developed in a rapid prototype-tcs!-revise cycle in close collaboration with science teams engaged in research. The early applications grew out of interactions with researchers in geology, oceanography and atmospheric physics and chemistry. It is now being used in many other fields such as physical chemistry, cellular biology, meteorology, and space plasmas. New users arc encour-aged to communicate their Needs for new tools and services, and recommendations for modifications of existing tools. Such communications have led to ad hoc collaborations which have been fruitful, result ing in a continuously evolving suite of applications useful across many disciplines. The early applicat ions were modeled upon the kind of graphs scientists have been using for many years with the added elements of high interactivity and rapid or animated renderings.

LinkWinds is a highly dynamic system. Each time a control is changed with a button-down mouse movement, upon the button-up condition, a message with its new value is broadcast to the applications that arc linked to it. These applicat ions respond immediately by redrawing their displays of the data. Most of the cent rols can be put into a "track" mode in which, as long as the button-down condition is sustained, a continuous stream of messages is broadcast as the cent rol is being moved, Depending upon the size of the data slices being dealt with, these actions can result in real time animation on the LinkWinds screen. Wc will at tempt to convey some sense of the dynamics of this man ipulat ion as wc dc.scribe some of the applications.

Figure 4 shows a typical session to explore upper atmospheric chemical data collected by the Microwave Limb Sounder (MIS) [14] currently in orbit aboard the Upper Atmosphere Research Satellite (UARS). The LinkWinds top-level menu appears in the upper left. From this menu, databases, tools and system options are selected. Short form data objects, with their single link buttons, arc just below the menu. In this case, the data displayed arc upper at mospheric ozone and water vapor. These data objects have been linked to the various applications on the screen, as indicated by the inscriptions in their upper left corners. The window entitled **Image1** displays a global slice of the data at a pressure of 21.54 mbars, as selected by **Slider1**, which is linked to it. **Slider1** is also linked to **Image2**, which shows the water vapor at the same pressure. **Slider1** permits the user to scan the full da(a set from the maximum to minimum pressures (or altitudes). The user can also switch to any of the three orthogonal axes and similarly scan them. **Image** can do a variety of things including displaying data slices along any of three orthogonal axes, automatically stretching colors between the minimum and maximum data values, and image magnification, rotation and scrolling,. It has three embedded controls in the form of a point indicator (cross-hair), line indicator (profile line) and region indicator (bounding box), The "rgb" button enables three selected slices to be composited in red, green and blue colors, respectively. The "Histo" and "Line" but (ens launch ancillary display tools, which arc also available from the main menu. A color legend is provided at the bottom. The southern hemisphere ozone hole is shown at the lower left of **Image1**, and a high value of the water vapor is seen in the corresponding location in **Image2**.

A scatter plot is brought to the screen to study correlations between the ozone and water vapor. A strong anti-correlation is clearly seen in **Scatter1**, in the lower left corner, where the points shown come from the region defined by the bounding box control embedded in **Image** and linked to **Scatter1**. In the text box of **Scatter1**, statistical quantities associated with the scattered data arc given. Other statistical options arc available through mcnu sclcction, and a full rangc of information may bc savcd to a file via a button. The bounding box in **Image** can bc rcsizcd or moved as dcsircd for comparison with othcr regions of the data, The ozone data arc displayed in **Globel** as a height field rendered on a sphere. The region of reduced ozone at the south pole, and adjacent regions of higher ozone, arc clearly seen. **Sliderl** also sets the pressure of this display, and the height scale is varied by a rotary dial on the left side of the window. An **XY (2-Axis) Rotator** control is linked to **Globel** to allow it to bc positioned as desired. A **Pan-Zoom** Slider is also available for this purpose. To enable rapid positioning of the globe, it is converted to a wire-frame rendering while being moved, All 3-dimensional figures such as the globe arc polygonal renderings. The "?" button on each of the applications provides a context-sensitive help window.

Figure 5 shows real time data from the Plasma Wave Subsystem (PWS) aboard the Galileo spacecraft, taken during the Earth 2 encounter in December 1992. St **reamPlot** in the upper left contains a spectrogram of the data, displaying frequency channel vertically versus time horizontally. It is essentially a strip-chart recorder in which new data enters on the right, and falls off on the left. The colors displayed give the intensity of the electric field, The magnetic field may similarly bc shown by clicking on the "B-field" button. Inter active color st retching may bc done on any of three frequency ranges, corresponding to the different instruments of PWS, to provide better contrast and simulate the different calibrations of the instruments. The data values corresponding to each color arc shown in the color bar below. Turning off the "color" opt ion gives a histogr am mode, in which each frequency is shown as a bar plot with adjustable gain. The pixel width corresponding to one time-step may also bc adjusted by buttons at the window's lower right, and the entire window may bc stretched to enlarge the plotting area.

in the upper right of Figure 5, the **StreamPlane** application displays data similar to the **StreamPlot**, but rendered in perspective with the height also determined by intensity of the field. The perspective is controlled by the **Pan-Zoom** and **2-Axis Rotator** tools below, while the height gain is adjusted by the dial to the left on **StreamPlane**. The Filter **Slider** has been used to limit the frequency range to better concentrate on lower frequency fixtures. Because the three-dimensional data takes longer to render, the number of time steps displayed in St rcaml'lane is less than on StreamPlot. In the lower left of the figure, the **St reamLine** applicat ion displays the most recent spectrum in white. For comparison with subsequent data, a spectrum may bc frozen by pressing a button, and will then bc shown in green. in addition, the bounding box on the spectrogram selects a spectral and temporal region. The spectrum is then averaged over the time region and displayed in a different color. This spectrum may also bc frozen or continuously updated. Both **StreamLine** and St rcan~I'lane can display either electric or magnetic field, and may bc interactively resized. Lastly, the St **reamClock** at bottom center gives both internal spacecraft and Universal times.

in addition to the tools show in the figures, the following arc available:
3-Axis Slider controls which slice of data, along any of three orthogonal axes, is displayed.
3-Axis Rotator rotates three dimensional applications such as Plane and Globe about each of the three orthogonal axes.
Color **Tool** provides interactive palette manipulation of data sets during a LinkWinds session. Individual data values may bc specially marked, or a range of data may bc ramped in color between two values. A variety of palettes, defined by the user, may bc substituted for the original one.
Contour permits the user to define one or more data values for which contours arc constructed on the linked display objects. The contour lines can bc colored and relocated as desired, A data distribution histogram is provided to guide the selection of the values.
Combine allows different slices of data, from either the same or different data sets, to bc combined into images using various mathematical functions. Currently, up to three slices of data may bc combined with the options listed in the pull-down menu. In addition, a calculator is available to input arbitrary functions of both slim and constants. An embedded histogram may bc used for interactive rebinning or renormalizat ion of the computed data.
Compare compares the behavior of each point in a data set with regard to a reference point, and displays the resulting image. For example, the "Sum Square" option determines the value at a given point by taking the data value, subtracting the data value at the reference point, and squaring, The squares arc then summed for every slice in the data set. The resulting values for every point arc then binned into byte values and plotted. Compare is thus a useful tool for examining spectral data.
Dat a Subset saves subsets of the displayed data from user selected regions. The data arc saved in HDF formal with

all pertinent metadata.

LinePlot displays the values along a straight line going completely through a data set parallel to any of the three axes. Plots are controlled by both tracking and frozen crosshairs, as well as the bounding box, in which case the average value over the defined region is displayed, 1a addition, because al I slices or levels are show, LinePlot is provided with red, green, and blue slides to be used to select up to three different slices.

Histogram displays the distribution of values on a slice, for up to three slices of data. By using two sliders available for each slice, interactive filtering and color stretching may be done to create composite RGB true or false color images and to enhance feature detection on single slices. Yellow lines indicate the maximum and minimum data values. The gain of the histograms may be adjusted 10 enable the view'iap, of bins with very low number counts.

]'lane renders an image in relief, with an accompanying height field. The image may be either a single slice or a three slice RGB composite. Onc or two data sets may be loaded; their names are given at the upper left of the window. If there is just onc data set, then its pixel values will also act as the height field. With two data sets, the first set determines the image while the second acts as the height. The gain of the height is interactively adjusted by a dial. The rendered image is three dimensional, and may be rotated, panned, and zoomed. The portion of the image rendered may be determined by the bounding box in the various image display objects. The application window may be resized, with the size of the rendered plane adjusting accordingly.

Polar renders either a north- or south-polar projection of the data with an accompanying height field. Otherwise, it behaves much like **Plane**.

OrthoView displays a volume rendering of all points in a three dimensional data set which lie between two selected values.

Profile displays the data values along a line drawn on a slice of data. The slice may be from any view axis of **Image**, or from the **Combine** or Compare tools. The line is draw with a pencil shaped cursor called from the menu of the drawing application, which must be linked into Profile.

3D Scat**ter** Plot plots points from three data sets, At every location in a slice, the data value of one set is plotted against the data value of others. A bounding box may be used as to control the region scattered. The points arc color coded for reverse location on the originating image. A variety of statistical information is available, and may be saved to a file.

TrackPixel gives numer ical information as cent rolled by the crosshair or bounding box in any of the image display objects. The upper pad shows the location of the crosshair and the data value at the selected point. The lower pad gives statistical information from the bounding box.

Animations of any of the displays arc supported by two tools, They arc Animator, which is tilnc-based, and **Frame** Animator, which is frame-based, For each, the user defines key frames with control settings and the animation tool dots "inbetweening" by interpolating intermediate settings for the controls. All of the controls used in the animation arc linked through the animators rather than directly to the display objects. in turn, the animator is linked to the display object, The **Frame** Animator is the simpler and allows defining only two kcy frames, and the number of inbetween frames desired, The time-based Animator allows any number of key frames to be defined, and the select ion of the time intervals between these frames. The inbetween interpolations can be along either linear or splint curves, and at a variety of frame rates including film and video. The saved frames can then be played back immediately or recorded for subsequent replay. Comprehensive time-based or space-based diagram arc provided snowing the values taken by each control guiding the animation.

FUTURE WORK

Active control of **remote** processes

With its MUSE subsystem, Link Winds can already exercise cent rol over another running LinkWinds process anywhere on the internet. It is a straightforward task to expand this capability to other kinds of processes, be they computations and computer simulations, or hardware devices capable of computer control such as scientific instruments or spacecraft. LinkWinds controls, specific to such processes, will be developed, extending the user interface to exercise the control functions with sliders, dials, displays, etc., as needed. lynx commands would then be broadcast over the networks to sockets on a remote server or servers designed to exercise specific devices.

Annotation **tools**

LinkWinds currently has an overlay system in which the user **can** provide a file containing a list of vector coordinates and have them rendered over the data, Wc will expand this capability by providing tools to allow the user to create these overlay files interactively. This will permit fully annotated overlays for data analysis, and the interactive preparation of figures for publication.

Support **for** additional **data** file formats

Until the advent of the real time data interface, research using LinkWinds was confined to relatively well-behaved and rectangularly gridded data sets. Essentially all of the general tools built for visualizing and analyzing data also concentrate on this category of data. While LinkWinds started, as any orderly development should, from the. most common circumstances, the development will move into more complex data analysis problem areas. There are major neglected categories of data that are generally quite common in NASA scientific research, and badly in need of tools to support their reformatting, exploration, and analysis. in these research areas, as in all areas of LinkWinds development, methodology and application building will be drive.n by real scientific problems. Several such areas are identified at this time.

(1) Data sets in which there are significant sources of error, both statistical and systematic. There are many areas of research in which errors in the measurements greatly impact their interpretation. These include all astronomics in which flux values are low, notably x-ray, gamma-ray, cosmic-ray and neutrino astronomics. Associated with each pixel of an image is an error. As the images are viewed, or processed, the errors must likewise be dealt with so that the images can be properly interpreted.

(2) Data sets which are ungridded samples, either sparse or numerous, from which it is desired to construct gridded data sets over extended regions, A notable example is in oceanography where ocean soundings yield vast quantities of information for isolated points. Tools are needed to construct maps for large regions by interpolating these measurements. This is often most easily done visually, but tools are scar-ce at this time. This need also arises in both atmospheric and fields and particles research.

(3) Disparate sized data sets which must be warped and/or co-registered for overlay or comparison, In single- or multi-disciplinary studies, data or images from various instruments are compared or combined for study. In general, all of these data sets are on different scales and in different orientations, and must be normalized to make the comparisons valid. Again, this is most rapidly and effectively done interactively and visually. Such situations are now prevalent in all areas of research.

Develop **User Applications Generator**

Currently the layout of all objects within the LinkWinds' windows is determined by a text file. The user can re-configure any of the windows by editing this file. LinkWinds also has a menu-selectable "redesign" mode which allows the user to interactively move any object in a window to a new location. When either the session or the redesign mode is ended, the new text file entries for the revised applications are saved enabling the user to subsequently add them to the layout file, We intend to expand this tool kit approach to allow users not only to move objects around within a window, but to take them off and throw them away, or add new ones from a provided catalog. in conjunction with the object selections, LinkWinds will generate a C-code source module to exercise their functionality. This code will be suitable for use as a template for the development of a full application. Initially, the user will be required to write the code which renders or displays the data in the places designated by the template. As experience is gained with this approach, it is expected that the rendering and display processes will also lend themselves to a limited catalog of processes selectable by the user.

Implementation on other platforms

There is a need to port LinkWinds to other platforms. in anticipation of this task, all functionality previously dependent upon the SGI GL-library has been converted to X Windows, with the exception of that used for rendering the data. The rendering functions of the SGI GL-library are contained in the OpenGL library [7] which is a strong contender for standardization as an X Windows 3-dimensional rendering widget basis. A number of personal computer and workstation manufacturers have licensed the library and plan to support it in future operating systems, This will make the porting of LinkWinds to any X Windows system straightforward, Meanwhile, efforts using third-party libraries to port SGI GL-library-based applications are under way.

Acknowledgements

The authors wish to acknowledge important contributions made by Mark Rubin and LinkWinds alumni Brian Beckman, Leo Blume, Bonnie Royal and Mitch Wade. The LinkWinds time-based animator was designed and implemented by our colleagues, Jim McLeod and Phil Mercurio of the San Diego Supercomputer Center. We want to thank several scientists who have used LinkWinds and provided us with significant and valuable feedback. They are Scott Bolton, Lee Elson and Andy Tran of the Jet Propulsion Laboratory, Kathi Beratan of tile University of Pittsburgh, Mike Botts and Dean Cutten of Marshall Space night Center, Bill Kurth of the University of lows, Charlie Rubin of Central Washington University and Joe Gatewood of Los Alamos National Laboratory. We extend our thanks to Joe Bredekamp and Glenn Mucklow of NASA, whose Applied information Systems Research Program makes this work possible. The research described in this publication was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

References

1. Brown, S. A., Folk, M., Goucher, G. and Rew, R., Software for Portable Scientific Data Management, *Computers in Physics*, 7, 2(h4ay/June 1993)304.

2. Donoho, A. W., Donoho, D.L. and Gasko, M., MacSpin: Dynamic Graphics on a Desktop Computer, *IEEE Computer Graphics & Applications*, *8*, 4 (July 1988), 51.

3. Dyer, D. S., A Dataflow Toolkit for Visualization, *IEEE Computer Graphics & Applications, 10*, 4 (July 199(J), 60.

4. Haeberli, P.E., ConMan: A Visual Programmin g Language for Interactive Graphics, *Computer Graphics*, 22,4 (August 1988), 103.

5. Handley, T.H., Jr. and Rubin, M.R., DataHub: Intelligent Distributed Science Data Management to Support Interactive Exploratory Analysis, Al AA, *Computing in Aerospace*, 9, (1993) (in press).

6. Kriz, R. D., PV-Wave Point & Click , *Pixel*, (July/August 1991), 28.

7. OpenGL Architecture Review Board, OpenGL Reference Manual, Actdison-Wesley Publishing Co., Reading, MA, 1992.

8. Rasure, J.R. and Williams, C. S., An Integrated Visual Language and Software Development Environment, *Journal of Visual Languages and Computing*, *2*, (1 991), 217.

9. Rettig, M., Nobody Reads Documentation, *CACM, 34*, (1991), 19.

10. Rew, R. and Davis, G., NetCDF: An Interface for Scientific Data Access, *IEEE Comp uter Graphics & Applications, 10*, 4 (July 1990), 76.

11. Schneiderman, B., Designing the User Interface, Addisiol~-Wesley Publishing Co., Reading MA, (1 992) 181.

12. Stapleton, L., A Sharper Edge, *Computer Graphics World*, (June 1993), 84.

13. Upson, C., Faulhaber, T., Jr., Kamins, D., Laidlaw, D., Schlegel, D., Vroom, J., Gurwitz, R., and van Dam, A., The Application Visualization System: A Computational Environment for Scientific Visualization, *IEEE Computer Graphics & Applications*, *9*, 4 (July 1989), 30.

14. Waters, J. W., Froidevaux, L., Read, W. G., Manney, G.L., Elson, L. S., Flower, D. A., Jarnot R.F. and Harwood, R, S., Stratospheric ClO and Ozone from the Microwave Limb Sounder on the Upper Atmosphere Research Satillite, *Nature*, 362, (1993) S97.

15. Wells, D. C., Greisen, E. W., and Harten, R.H., FITS: A Flexible Image Transport System, *Astron. Astrophys. Suppl.*, 44, (1981) 363-370.

16. Witten, M., Adding it Up, *Sun World*, (October 1992).

About the Authors

Al .I.AN S. **(BUD) JACOBSON is** a Scnior Research Scicntist in the Earth and Space Scicnccs Division of the Jet Propulsion I .aboratory. His research intcrcsts span both the physical scicnccs and computer science and includc high energy astrophysics, interdisciplinary scicncc, visual data analysis mcthods and intcractive computer graphics.

ANDREW L. BERKIN is a Membcr of the Technical Staff in the Earth and Space Sciences Division of the Jet Propulsion Laboratory. 1 lis research interests include gcncral relativity, cosmology and the application] of computer graphics to the analysis of scicntific data.

MARTIN ORTON is a senior softwarc systcms enginccr in the Earth and Space Scienccs Division of the Jet l'repulsion Laboratory. His research intcrcsts arc intcrpretive languagcs and object-oIicIItcd softwarc tcchniqucs.

Authors' **Present Address:** Earth and Space Scicnccs Division, M/S 183-501, Jet l'repulsion Laboratory, 4800 Oak Grove Drivc, Pasadena, CA 91109, budj@apcx.jpl.nasa.gov, berkin@krazy.jpl.nasa.gov and martin@ignatz.jpl.nasa.gov.

Figure Captions

Figure 1: The short and long form of the LinkWinds data object

Figure 2. A slider, an examplc of a simplc LinkWinds control object,

Figure 3: An example of a LinkWinds display object, This particular example has control objects embedded in it

Figure 4: LinkWinds session to explore upper atmospheric ozone and water vapor measured by the Microwave Limb Sounder aboard UARS.

Figure S: Plasma wave data taken by PWS aboard Galileo during the Earth ? encounter in December 1992 and displayed in real time.

Figure 1. Short and long data object forms

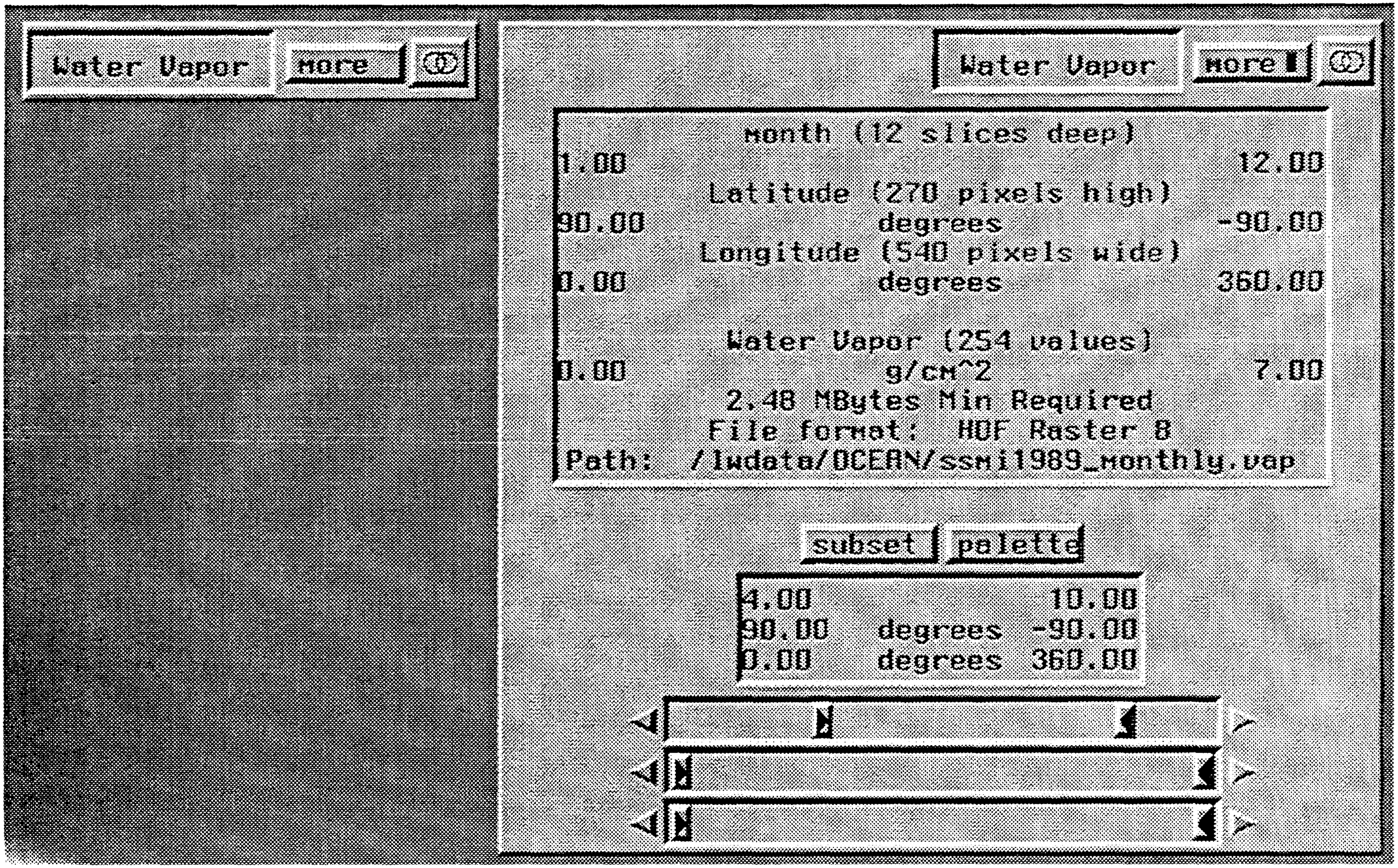


Figure 2. A control object

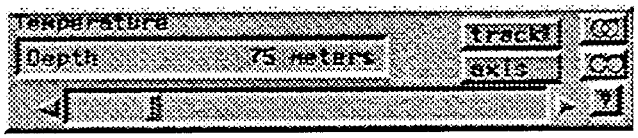
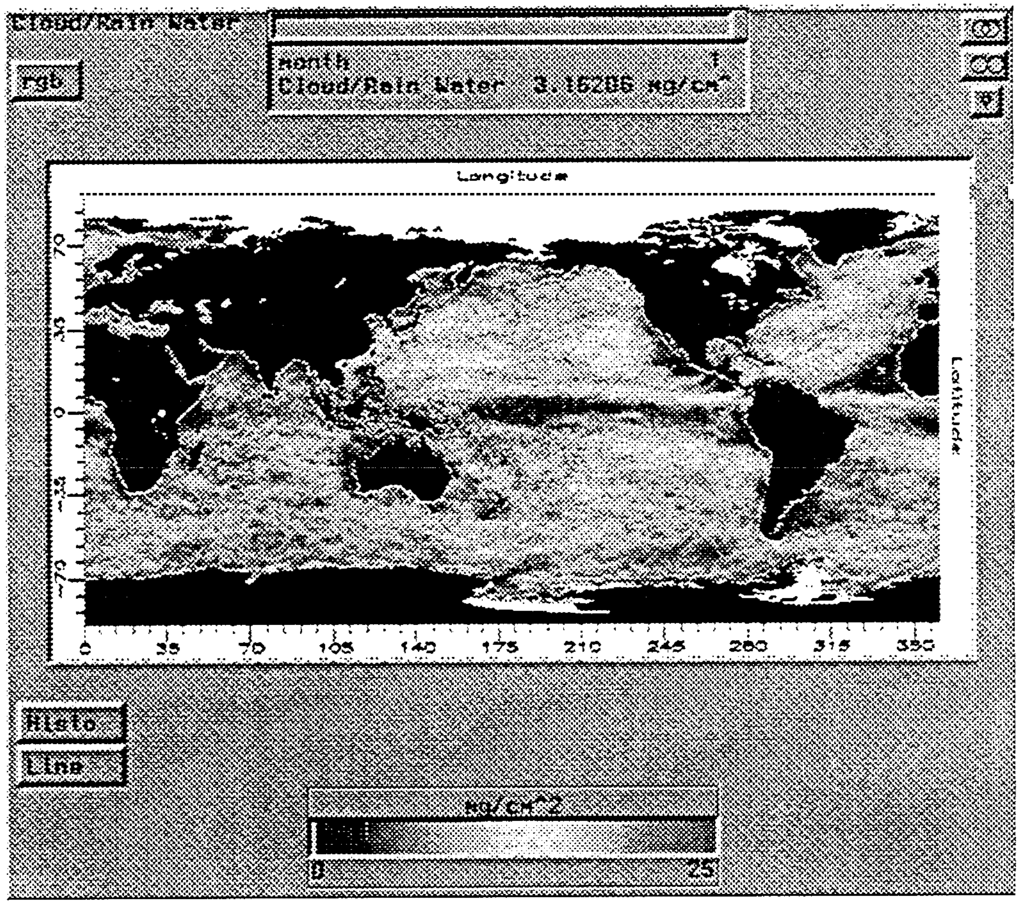


Figure 3. A display object with controllers



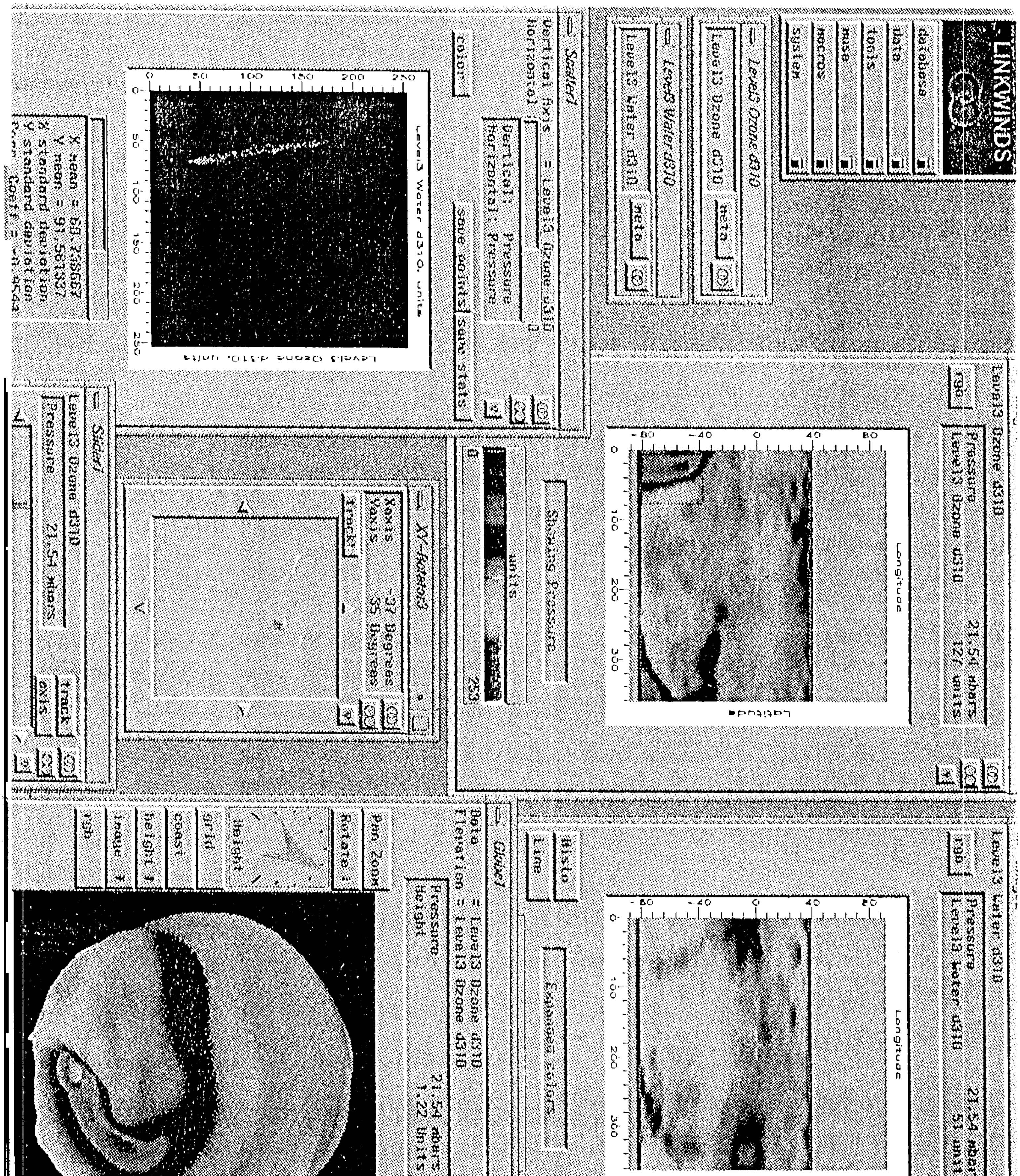


Figure 4: LinkWinds session to explore upper atmospheric ozone and water vapor measured by the Microwave Limb Sounder aboard UARS.

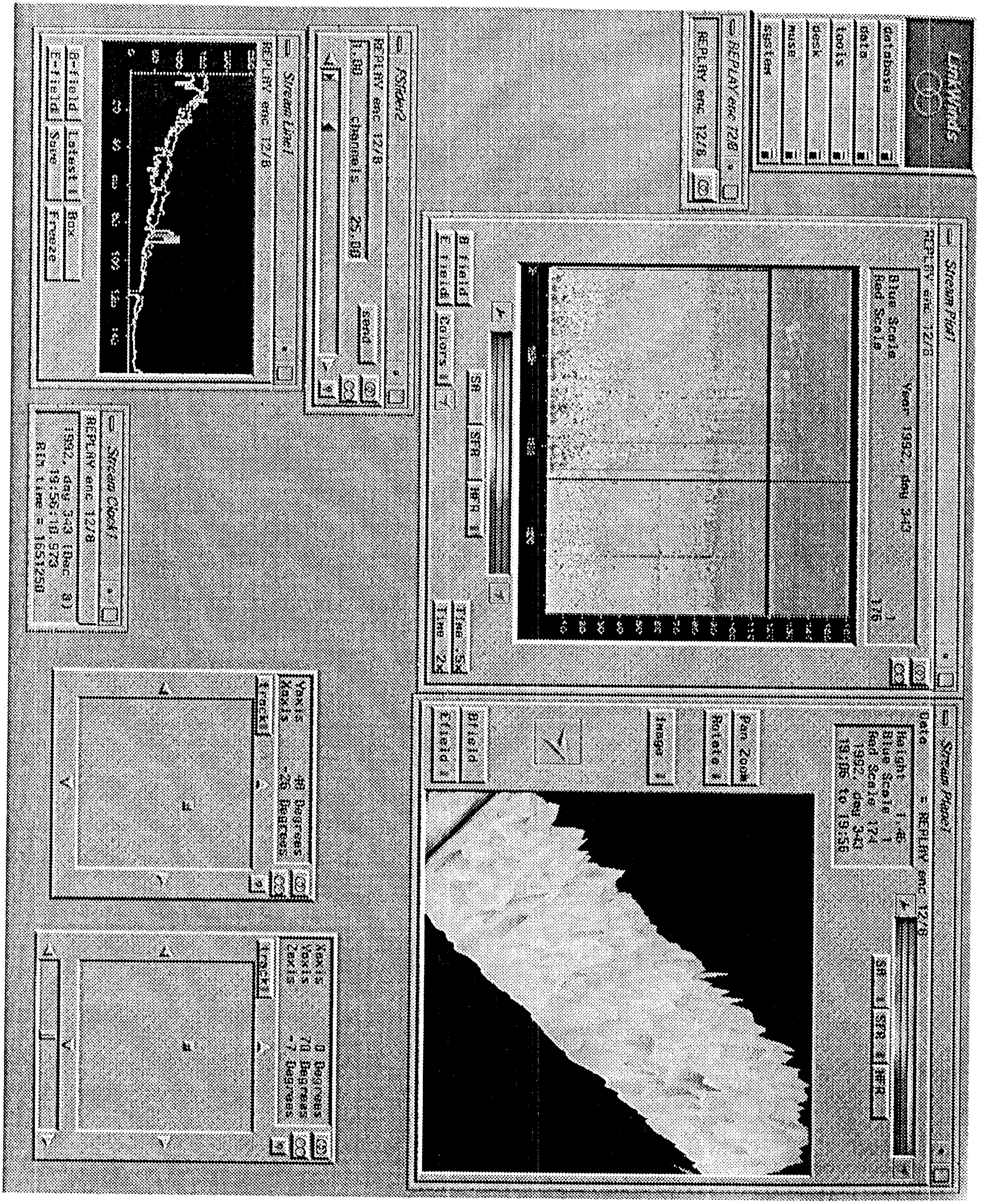


Figure 5: Plasma wave data taken by PWS aboard Galileo [illegible]